PROBLEM STATEMENT

MScFE Capstone Project

Student Group 11196

Kenneth Kariuki kenkarish1935@gmail.com
Davide Freni devide.freni@gmail.com
Vicky Singh ivickysingh@yahoo.com

September 21th, 2025

Project Track

8. Machine Learning (Deep) Investment Strategies, specifically Topic 2: Machine Learning (Deep) Application to Portfolio Allocation

This project follows the practical track, focusing on machine learning (deep learning) investment strategies.

Problem statement

The goal of this project is to move beyond traditional portfolio strategies, like Modern Portfolio Theory (Markowitz 1952), and focus on what actually works in practice. The proposed approach blends deep learning, classic machine learning, and reinforcement learning—each bringing something unique to the table.

We use LSTM networks to get a sense of how asset prices might evolve (Hochreiter and Schmidhuber 1997), XGBoost for more granular return predictions (Chen and Guestrin 2016), and reinforcement learning agents that can adapt allocation decisions as new data comes in (Moody and Saffell 2001). We benchmark these against tried-and-true models like random forests (Breiman 2001), and we never shy away from putting our ideas up against established strategies from both academia and industry.

The analysis uses a portfolio of nine highly liquid instruments:

- Stocks: Apple (AAPL), Microsoft (MSFT), Amazon (AMZN), NVIDIA (NVDA), Google (GOOGL)
- ETFs: SPDR S&P 500 ETF (SPY), Invesco QQQ Trust (QQQ)
- Cryptocurrencies: Bitcoin (BTC-USD), Ethereum (ETH-USD)

We are working with a mix of nine highly liquid assets—because, let's face it, liquidity is non-negotiable when you actually have to execute these trades (Pastor and Stambaugh 2003). The data will cover the time from September 2018 to September 2025, which includes a lot of market ups and downs. We evaluate how effectively these models work in real life, not just in theory, throughout both bull and bear cycles.

Here's how we've structured the project, step by step:

- **Data Collection & Prep**: First, we gather daily prices for all nine assets, making sure the data is clean and consistent. This part is tedious but absolutely critical—bad data leads to bad models (Lo and MacKinlay 1990).
- **Feature Engineering**: we pull in the usual suspects—lagged returns, rolling volatility, and RSI—plus the VIX to capture market mood swings (Fama and French 1989; Whaley 2000). Over the years, we've found that combining technical and macro indicators gives models a fighting chance.
- **Model Development:** The data preparation process leads us to train different models, which include reinforcement learning, LSTM, XGBoost, and random forest. The separate methods of each model generate unique investment suggestions for the upcoming trading day, which enables us to establish multiple investment distribution plans.
- **Backtesting:** The models receive new unseen data for backtesting to evaluate their actual market performance potential (Bailey et al. 2014). The portfolio receives daily updates through a simulated trading process, which produces results that match real-world market behavior.

- **Performance Assessment**: The evaluation process through performance assessment depends on an equal-weight benchmark, which serves as a neutral reference point to evaluate each strategy. Our evaluation system assesses both the risk protection and the reward-to-risk balance of investment strategies. The evaluation system uses established performance metrics, which include Sharpe (Sharpe 1966), Sortino (Sortino and Van der Meer 1991), and Calmar ratios (Young 1991) to measure maximum drawdown. The evaluation process focuses equally on statistical performance indicators and market volatility response of each model. The ability of a system to handle stressful situations proves more valuable than its minor advantages during a normal market.
- **Risk and Robustness Analysis**: The portfolio construction process can incorporate multiple layers of risk control to choose from. and we'll evaluate which are the best to be applied. Position sizes are scaled in relation to realized volatility (Hull 2018), leverage is capped to prevent excessive exposure, and stop-loss as well as take-profit thresholds are determined using average true range (Chande 1995). The system tests its durability through historical crisis period stress scenarios (Barro and Ursúa 2008) and Monte Carlo simulations to evaluate its resistance against extreme tail events (Glasserman 2004). Liquidity considerations are explicitly built into the framework (Pastor and Stambaugh 2003), and robustness is further validated across alternative datasets and additional asset classes, reducing the likelihood that results hinge on any single market environment. Position sizing is systematically adjusted in response to realized volatility; leverage is constrained; and stop-loss as well as take-profit rules are implemented on the basis of average true range (ATR). Stress-testing basically means running through past crisis scenarios to see how things hold up, while Monte Carlo methods help us explore those really rare, extreme events that might not have happened before. We also make sure to include liquidity limits in our tests and double-check our results using different datasets and a variety of asset types.
- **Concluding Remarks**: Our professional experience has taught us that no model should be regarded as infallible. Even if backtests look really good statistically, the real test is how the model performs in situations that don't match our expectations (Taleb 2007). It is in such circumstances that methodological rigor, collective expertise, and a disciplined degree of skepticism serve as indispensable safeguards.

Goals and Objectives

This study's main goal is to create and assess an effective dynamic portfolio allocation strategy using a range of models. Our approach involves several key objectives:

- Initially, we will build and train models such as LSTM, XGBoost, Random Forest, and potentially others that are capable of suggesting daily asset weights for a diversified multi-asset portfolio.
- Second, to develop a set of characteristics that make sense to improve our models by mixing historical asset data with a macroeconomic indicator such as VIX.
- Third, to set up a backtesting system that can pretty closely mimic how the strategy would have performed in the past, including realistic transaction costs.
- Finally, we want to compare how this active approach stacks up against a simple, equally weighted benchmark to see if it really helps improve risk-adjusted returns.

Code Implementation Plan

Tech Stack: Pandas, numpy, yahoo finance, scikit-learn, keras, tensorflow, XGBoost, TF-Agents, Matplotlib,

Step 1: Data collection and preprocessing:

Function: get_data(tickers, start_date, end_date)

- Select assets tickets and time frame
- Use yahoo finance to download the data
- Put "close" prices into a dataframe using pandas
- Handle missing values, duplicates, outliers, etc. preprocessing tasks
- Output: training and test dataset

Step 2: Feature Engineering

Function: create_features(data)

- Calculate daily returns
- For each of the assets, calculate lagged returns, 20-day rolling volatility and 14-day relative strength index (RSI).
- Include external market data as a proxy for systemic volatility.
- Normalization using Min-Max Scaler
- Output: scaled training and test datasets

Step 3: Model Training

Function: train_model(train_scaled_dataset, train_return_dataset)

- The asset with the highest next-day return is prediction target
- Implement LSTM, XgBoost and Random forest using engineered features
- Output: trained LSTM, XGBoost and Random forest model

Function: train_RLagent(train_scaled_dataset, train_return_dataset)

- Model market returns, portfolio returns and market dynamics using a custom environment
- Develop a PPO agent using actor and critic architectures
- During training, agents interact with the environment and optimize their policy for maximum returns
- Output: Trained RL agent

Step 4: Backtesting

Function: run_backtest(test_features, test_returns, models)

- Create a portfolio tracker for each agent and a baseline strategy with equal weights.
- Generate daily portfolio allocation for each model
- For supervised models, predictions will be obtained using predict(), while for RL agent through its policy
- Calculate daily returns using allocations of each model to the actual returns.
- For all the strategies, put daily returns in one dataframe.

Step 5: Evaluation

Function: evaluate_performance(all_returns)

- For every strategy, calculate key metrics
    - Cumulative returns
    - Sharpe ratio
    - Sortino ratio
    - Maximum Drawdown
- Comparative analysis using a summary table of results.

References

Bailey, David H., et al. The Probability of Backtest Overfitting. Journal of Computational Finance, vol. 20, no. 4, 2014, pp. 39–69.

Barro, Robert J., and José F. Ursúa. “Macroeconomic Crises Since 1870.” Brookings Papers on Economic Activity, Spring 2008, pp. 255–335.

Breiman, Leo. “Random Forests.” Machine Learning, vol. 45, no. 1, 2001, pp. 5–32.

Chande, Tushar S. The New Technical Trader. Wiley, 1995.

Chen, Tianqi, and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

Fama, Eugene F., and Kenneth R. French. “Business Conditions and Expected Returns on Stocks and Bonds.” Journal of Financial Economics, vol. 25, no. 1, 1989, pp. 23–49.

Glasserman, Paul. Monte Carlo Methods in Financial Engineering. Springer, 2004.

Hochreiter, Sepp, and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation, vol. 9, no. 8, 1997, pp. 1735–1780.

Hull, John. Risk Management and Financial Institutions. Wiley, 2018.

Lo, Andrew W., and A. Craig MacKinlay. A Non-Random Walk Down Wall Street. Princeton UP, 1990.

Markowitz, Harry. “Portfolio Selection.” The Journal of Finance, vol. 7, no. 1, 1952, pp. 77–91.

Moody, John, and Matthew Saffell. “Learning to Trade via Direct Reinforcement.” IEEE Transactions on Neural Networks, vol. 12, no. 4, 2001, pp. 875–889.

Pastor, Lubos, and Robert F. Stambaugh. “Liquidity Risk and Expected Stock Returns.” Journal of Political Economy, vol. 111, no. 3, 2003, pp. 642–685.

Sharpe, William F. “Mutual Fund Performance.” The Journal of Business, vol. 39, no. 1, 1966, pp. 119–138.

Sortino, Frank A., and Robert van der Meer. “Downside Risk.” Journal of Portfolio Management, vol. 17, no. 4, 1991, pp. 27–31.

Taleb, Nassim Nicholas. The Black Swan: The Impact of the Highly Improbable. Random House, 2007.

Whaley, Robert E. “The Investor Fear Gauge.” The Journal of Portfolio Management, vol. 26, no. 3, 2000, pp. 12–17.

Young, Terry W. “Calmar Ratio: A Smoother Tool.” Futures Magazine, Apr. 1991, pp. 40–41.